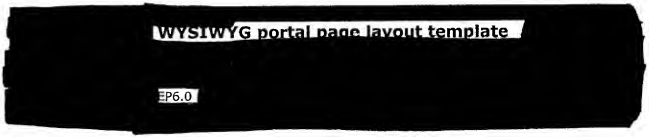
EXHIBIT A



The Concept

Using Template files with place holders

The provided solution is based on layout template files.
Those files use dedicated tags to set place holders for containers which contain the content components.
At run-time the placeholders will be replaced with the actual components presenting their content while at design-time the placeholders will present content components representation to simulate the run-time presentation.
The page general structure is determined by using html tags (like html <table>).
The template files can contain any other html elements or additional code, which will be rendered and presented both at run-time and design-time.

One file for both run-time and design-time

Using the same layout template file for both run-time (presenting the actual page to the user) and design-time (editing the page content), provide create-once-use-twice solution and enables easy maintenance.
This architecture ensures presenting exactly same page appearance both at run-time and at design-time.

The Implementation

Jsp layout template file with dedicated tag library tags

The implementation uses jsp files as the layout templates. Such jsp file can contain any html, java, tag-lib or other jsp complying content, to be included in the portal page.
The content components containers place holders are marked with a dedicated tag library, using the <lyt:container> tag. Such tag, located in the jsp, marks a place for a column of content components (iViews).

Run-time and Design-time rendering

The provided solution uses different <lyt:container> tag handler implementations for run-time and design-time.
The run-time tag handler includes the content components' run-time generated content.
The design-time tag handler paints a schematic run-time-look-alike representation of each content component with code enabling edit the page (adding, removing, re-locating components).
All other jsp content is rendered the same.